Fondamenti di Internet e Reti

Antonio Capone, Achille Pattavina,
Francesco Musumeci, Matteo Cesana

POLITECNICO
MILANO 1863

Scripting con Python

Antonio Capone, Achille Pattavina,
Francesco Musumeci, Matteo Cesana

Attività di laboratorio: Versioni software

Gli esempi mostrati a lezione utilizzano:

- Python versione 3.4
- PyCharm IDE Education Edition
    - include python 3.4

Sono entrambi pre-installati nella macchina virtuale

In alternativa potete installare Python e PyCharm sul vostro computer (+ libreria matplotlib)

Tempo di risposta di un server HTTP

- Valutare il tempo di risposta del server di Google per scaricare l'home page

```
1  import requests
2
3  r = requests.get('http://www.google.com')
4  print('Tempo di Risposta:', r.elapsed.microseconds / 1000, 'ms')
```

- Ripetere la misura 10 volte

```
1  import requests
2
3  for _ in range(10):
4      r = requests.get('http://www.google.com')
5      print('Tempo di Risposta:', r.elapsed.microseconds / 1000, 'ms')
```

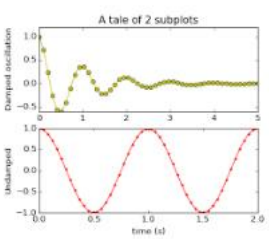
- Come possiamo calcolare minimo, media e massimo?

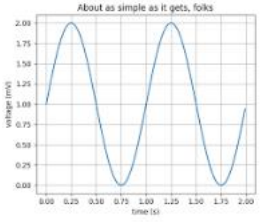
Calcolo minimo, media e massimo

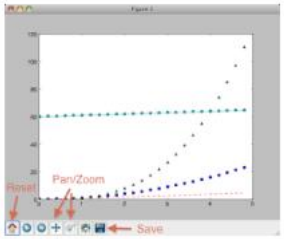
```python
import requests

tempi = []
for _ in range(10):
    r = requests.get('http://www.google.com')
    tempi.append(r.elapsed.microseconds / 1000)

print('Tempo di Risposta - MIN:', min(tempi))
print('Tempo di Risposta - AVG:', sum(tempi)/len(tempi))
print('max', max(tempi))
```

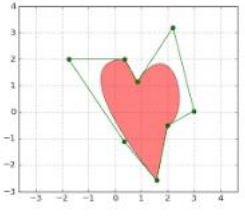
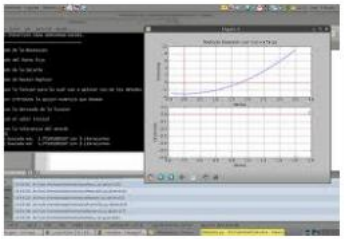
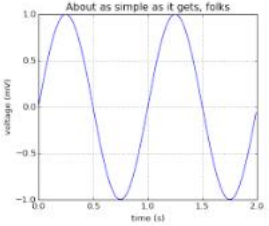
- Come possiamo rappresentare graficamente i risultati?

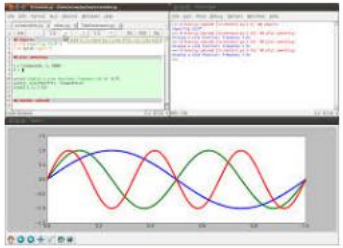
matplotlib

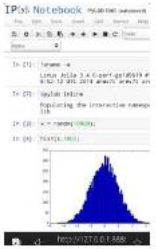


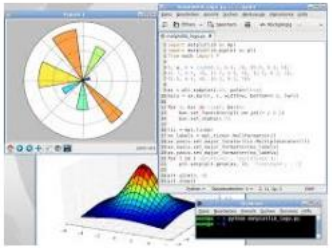
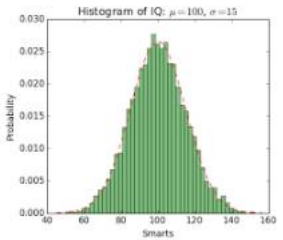


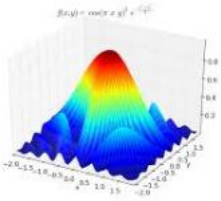
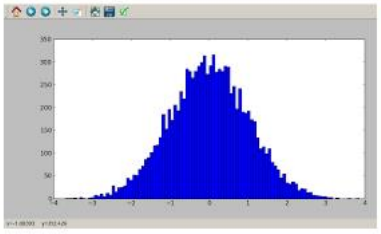
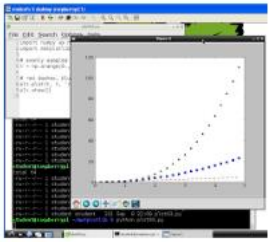
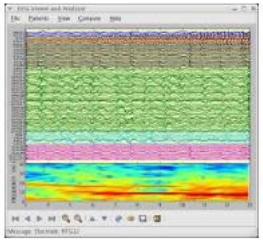
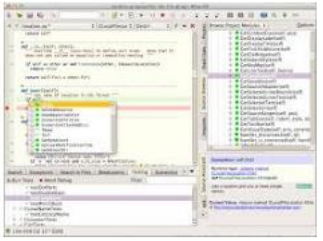
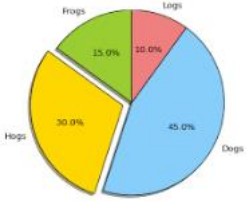


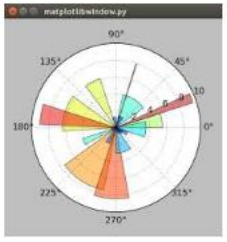










http://matplotlib.org/examples/

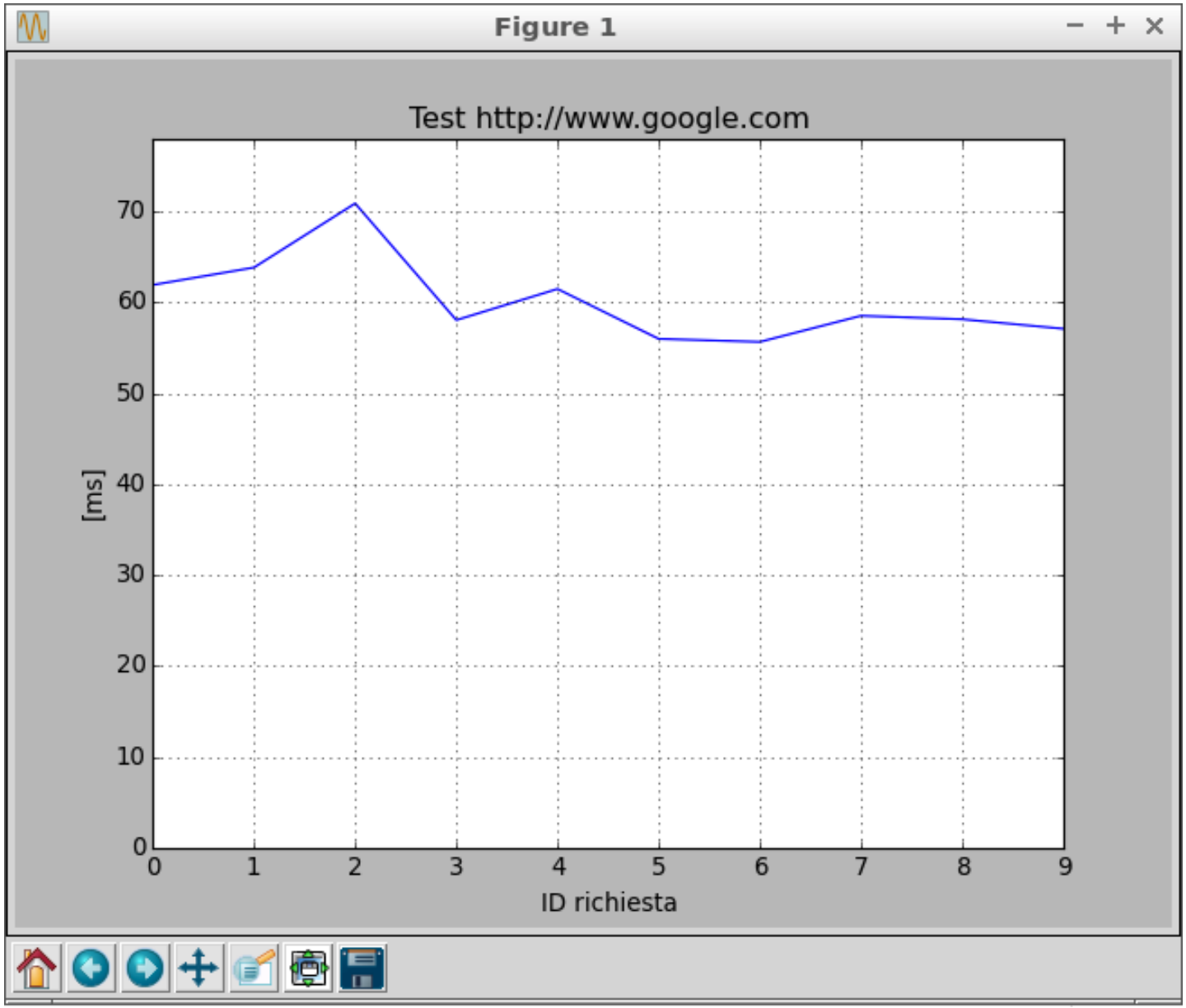
Grafici con Python (1)

```
1   import requests
2   import matplotlib
3   matplotlib.use('TkAgg')
4   import matplotlib.pyplot as plt
5
6   tempi = []
7   for _ in range(10):
8       r = requests.get('http://www.google.com')
9       tempi.append(r.elapsed.microseconds / 1000)
10
11  print('Tempo di Risposta - MIN:', min(tempi))
12  print('Tempo di Risposta - AVG:', sum(tempi)/len(tempi))
13  print('Tempo di Risposta - MAX:', max(tempi))
14
15  plt.figure()
16  plt.plot(tempi)
17  plt.ylim([0, max(tempi)])
18  plt.show()
```

Grafici con Python (2)

```
1   import requests
2   import matplotlib.pyplot as plt
3   tempi = []
4   for _ in range(10):
5       r = requests.get('http://www.google.com')
6       tempi.append(r.elapsed.microseconds / 1000)
7
8   print('Tempo di Risposta - MIN:', min(tempi))
9   print('Tempo di Risposta - AVG:', sum(tempi)/len(tempi))
10  print('Tempo di Risposta - MAX', max(tempi))
11
12  plt.figure()
13  plt.plot(tempi)
14  plt.ylim([0, 1.1*max(tempi)])
15  plt.xlabel('ID richiesta')
16  plt.ylabel('[ms]')
17  plt.title('Test http://www.google.com')
18  plt.grid()
19  plt.show()
```

Grafici con Python (3)



Tempo di risposta di server HTTP multipli

```python
import requests

siti = ['http://www.gazzetta.it', 'http://www.netflix.com', 'http://www.facebook.com']
for url in siti:
    print('Test', url)
    tempi = []
    for _ in range(10):
        r = requests.get(url)
        tempi.append(r.elapsed.microseconds/1000)
    print('Tempo di Risposta - MIN:', min(tempi))
    print('Tempo di Risposta - AVG:', sum(tempi)/len(tempi))
    print('Tempo di Risposta - MAX', max(tempi))
```

```python
for ID_url in range(len(siti)):
    print('test sito #', ID_url)
    r = requests.get(siti[ID_url])
```

```python
for ID_url, url in enumerate(siti):
    print('Test sito #', ID_url)
    r = requests.get(url)
```

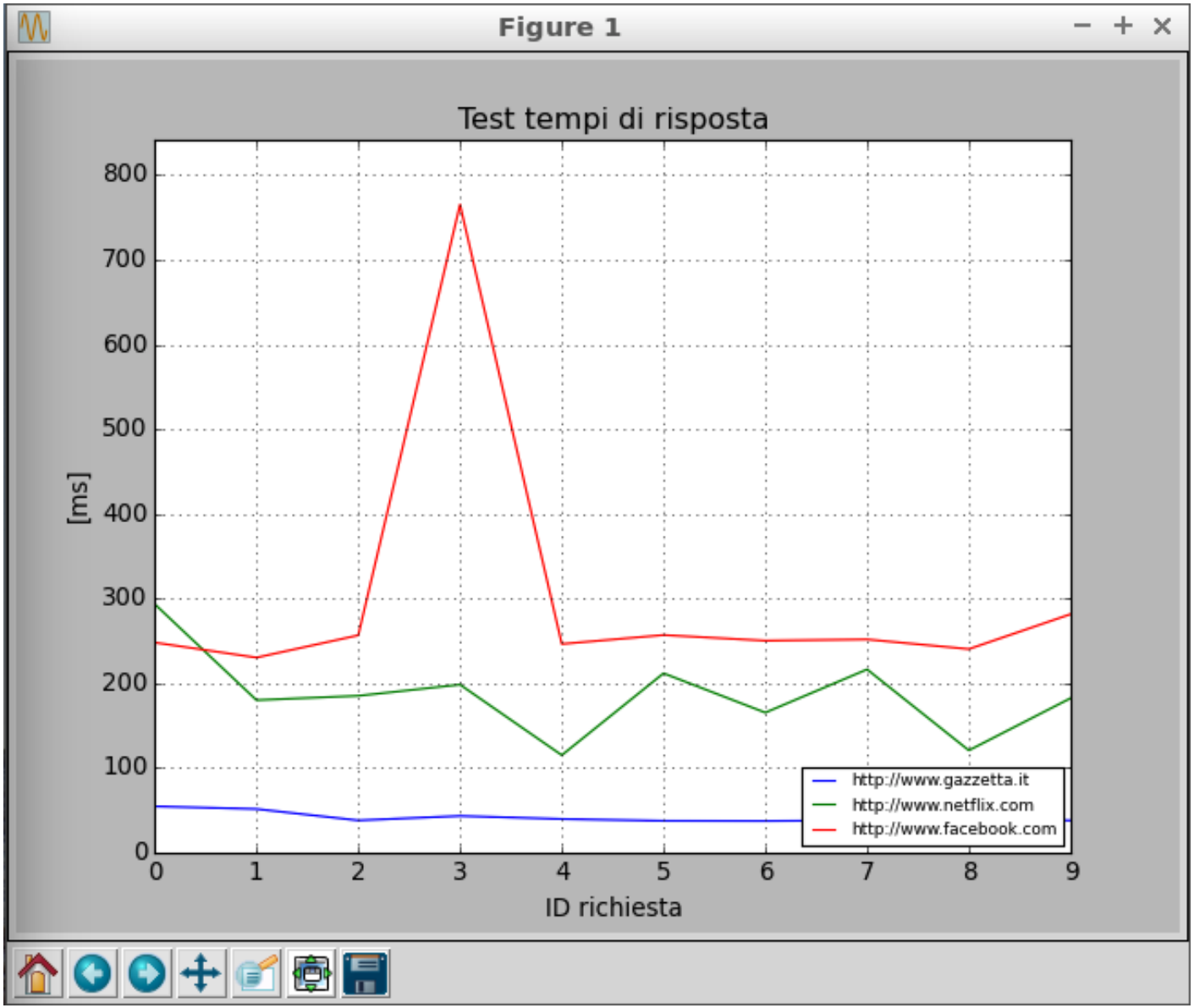
VS

```python
for url in siti:
    r = requests.get(url)
```

Grafici con server HTTP multipli

```
1   import requests
2   import matplotlib.pyplot as plt
3   m = 0 # massimo tra i massimi
4   plt.figure()
5   siti = ['http://www.gazzetta.it', 'http://www.netflix.com', 'http://www.facebook.com']
6   for url in siti:
7       print('Test', url)
8       tempi = []
9       for _ in range(10):
10          r = requests.get(url)
11          tempi.append(r.elapsed.microseconds/1000)
12      plt.plot(tempi, label=url)
13      print('Tempo di Risposta - MIN:', min(tempi))
14      print('Tempo di Risposta - AVG:', sum(tempi)/len(tempi))
15      print('Tempo di Risposta - MAX', max(tempi))
16      m = max([m, max(tempi)])  # ricalcolo il massimo tra i massimi
17
18  plt.ylim([0, 1.1*m])
19  plt.xlabel('ID richiesta')
20  plt.ylabel('[ms]')
21  plt.title('Test tempi di risposta')
22  plt.legend(loc='lower right', fontsize=8)
23  plt.grid()
24  plt.show()
```

Grafici con server HTTP multipli (2)



Esercizio

Scrivere uno script che stampi il nome della pagina col miglior tempo di risposta medio tra **2** siti Internet. Ripetere con **6** siti Internet.

- Numero di richieste = 10
- Siti internet:

A

```
http://www.google.com
http://www.youtube.com
http://www.polimi.it
http://www.wikipedia.org
http://www.amazon.com
http://www.twitter.com
```

B

Hint (per la parte B):

LISTA.index(x) ritorna la posizione dell'elemento x nella lista *LIST*.
È l'inverso dell'accesso alla lista tramite posizione

```
L = [1, 5, 20, 4]
print(L.index(5)) # STAMPA 1
```

```
L = [1, 5, 20, 4]
print(L[1]) # STAMPA 5
```

NB: gli indici in Python iniziano da 0!

Soluzione parte A

```python
import requests

tempi1 = []
for _ in range(5):
    r = requests.get('http://www.google.com')
    tempi1.append(r.elapsed.microseconds/1000)
avg1 = sum(tempi1)/len(tempi1)

tempi2 = []
for _ in range(5):
    r = requests.get('http://www.youtube.com')
    tempi2.append(r.elapsed.microseconds/1000)
avg2 = sum(tempi2)/len(tempi2)

if avg1 < avg2:
    print('http://www.google.com')
else:
    print('http://www.youtube.com')
```

Soluzione parte B

```python
import requests

siti = ['http://www.google.com', 'http://www.youtube.com', 'http://www.polimi.it',
        'http://www.wikipedia.org', 'http://www.amazon.com', 'http://www.twitter.com']
avg = []
for url in siti:
    print('Test', url)
    tempi = []
    for _ in range(10):
        r = requests.get(url)
        tempi.append(r.elapsed.microseconds/1000)
    print('avg', sum(tempi)/len(tempi))
    avg.append(sum(tempi)/len(tempi))

print(siti[avg.index(min(avg))], min(avg))
```

```python
siti = ['SITO1','SITO2','SITO3']
avg = [150, 90, 200]
minimo = min(avg) # =  90
indice_minimo = avg.index(min(avg)) # = 1
indice_elemento_90 = avg.index(90) # = 1
sito_corrispondente = siti[avg.index(min(avg))] #  =  siti[1] = 'SITO2'
```